

IN THE CLAIMS:

The text of all pending claims, (including withdrawn claims) is set forth below. Cancelled and not entered claims are indicated with claim number and status only. The claims as listed below show added text with underlining and deleted text with ~~strikethrough~~. The status of each claim is indicated with one of (original), (currently amended), (cancelled), (withdrawn), (new), (previously presented), or (not entered).

1. (previously presented) A file replication system having a plurality of nodes connected to a network, files being distributed to the nodes, wherein

a first node of the nodes comprises:

a first token managing portion giving access permission for a file within the first node when no other node has update permission and otherwise issuing a notification of a permitted node that has update permission for the file in response to an access request in the first node, and

an IO request intercepting portion accepting an access to the file, the access taking place in the first node when said IO request intercepting portion is capable of acquiring the access permission, asking said first token managing portion to acquire the access permission against the access request, and asking the permitted node that has update permission for the file to access to the file when said first token managing portion is not capable of acquiring the access permission, and

a second node comprises a second token managing portion notifying a requesting node that requests the access permission for the file of the permitted node that has the update permission for the file as a response message.

2. (previously presented) A node, connected to at least one other node through a network, every node having a copy of files synchronized with files of other nodes for high availability and high performance to provide a replicated file system, comprising:

a token managing portion managing an access request for a file; and

an IO request intercepting portion asking said token managing portion to acquire access permission for the file against an access request to the file in said node, said token managing portion giving access permission when no other node has update permission for the file and said token managing portion notifying said IO request intercepting portion of another node that has the update permission when the other node has the update permission for the file, in response to the access request of said IO request intercepting portion, said IO request intercepting portion accessing the file in said node when the IO request intercepting portion is capable of acquiring

the access permission and said IO request intercepting portion asking the other node that has the update permission to access the file instead of accessing the file in said node when said IO request intercepting portion is not capable of acquiring the access permission.

3. (previously presented) The node according to claim 2, further comprising:
a system structure managing portion performing a restoration process of data of a file of the node when it is newly joined to a system,

wherein while said system structure managing portion is restoring the file, when an access request for the file takes place in the node, said IO request intercepting portion asks another node that shares the file to access the file.

4. (previously presented) The node according to claim 2, further comprising:
a changed data notifying portion propagating an updated content of the file to other node along with information that represents a dependent relationship with another update; and
a received data processing portion reflecting the updated content to the file while assuring an order of the update based on the dependency relationship.

5. (previously presented) The node according to claim 4, further comprising:
a system state information portion storing information about propagation mode of an updated content for each of at least one file,
wherein said changed data notifying portion propagates the update content based on information queued in said system information portion.

6. (previously presented) The node according to claim 5, wherein the propagation mode is one of a synchronous mode in which it is assured that the updated content is propagated to all the nodes that share the file, a semi-synchronous mode in which it is assured that the updated content is propagated to the majority of nodes that share the file, and an asynchronous mode in which it is not acknowledged that the updated content is propagated to the nodes that share the file.

7. (previously presented) The node according to claim 4, wherein said system state information storing portion keeps information about each node that shares at least one file for each file.

8. (previously presented) A node, connected to at least one other node through a network, every node having a copy of files synchronized with files of other nodes for high availability and high performance to provide a replicated file system, comprising:

a token managing portion asking another node to acquire an access permission for a file against an access request for the file in said node; and

an IO request intercepting portion accepting an access request for a file in said node, asking said token managing portion to acquire the access permission for the file against the access request to the file in said node, said token managing portion giving access permission when no other node has update permission for the file and otherwise notifying said IO request intercepting portion of another node that has the update permission for the file, said IO request intercepting portion accessing the file in said node when the IO request intercepting portion is capable of acquiring the access permission and asking the other node that has the update permission for the file to access the file according to the access request instead of accessing the file in said node when said token managing portion is not capable of acquiring the access permission for the file.

9. (cancelled)

10. (previously presented) A file replication system having a plurality of nodes connected to a network, files being distributed to the nodes, wherein

a first node of the nodes comprises:

first token managing means for giving access permission for a file within the first node when no other node has update permission and otherwise issuing a notification of a permitted node that has update permission for the file in response to an access request in the first node, and

IO request intercepting means for accepting an access to the file, the access taking place in the first node when said IO request intercepting portion is capable of acquiring the access permission, asking said first token managing means to acquire the access permission against the access request, and asking the permitted node that has update permission for the file to access to the file when said first token managing means is not capable of acquiring the access permission, and

a second node comprises second token managing means for notifying a requesting node that requests the access permission for the file of the permitted node that has the update permission for the file as a response message.

11. (previously presented) A node, connected to at least one other node through a network, every node having a copy of files synchronized with files of other nodes for high availability and high performance to provide a replicated file system, comprising:

token managing means for managing an access request for a file; and

IO request intercepting means for asking said token managing means to acquire an access permission for the file in response to an access request to the file in said node, said token managing means giving access permission when no other node has update permission for the file and said token managing portion notifying said IO request intercepting means of another node that has the update permission when the other node has the update permission for the file, in response to the access request of said IO request intercepting means, said IO request intercepting portion accessing the file in said node when the IO request intercepting portion is capable of acquiring the access permission and said IO request intercepting means asking the other node that has the update permission to access the file instead of accessing the file in said node when said IO request intercepting means is not capable of acquiring the access permission.

12. (previously presented) A node, connected to at least one other node through a network, every node having a copy of files synchronized with files of other nodes for high availability and high performance to provide a replicated file system, comprising:

token managing means for asking another node to acquire an access permission for a file against an access request for the file in said node; and

IO request intercepting means for accepting an access request for a file in said node, asking said token managing means to acquire the access permission for the file against the access request to the file in said node, said token managing portion giving access permission when no other node has update permission for the file and otherwise notifying said IO request intercepting portion of another node that has the update permission for the file, said IO request intercepting portion accessing the file in said node when the IO request intercepting portion is capable of acquiring the access permission and asking the other node that has the update permission for the file to access the file according to the access request instead of accessing the file in said node when said token managing means is not capable of acquiring the access permission for the file.

13. (cancelled)

14. (previously presented) A file replication control method for a system having a plurality of nodes connected to a network, files being distributed to the nodes, comprising:
causing an access requesting node to access a file of the access requesting node itself when the access requesting node has the latest data of the file and has or is able to obtain access permission from another node having update permission for the file; and
asking the other node to access the file when the other node has the update permission for the file which is given to only one node at a time.

15. (cancelled)

16. (previously presented) The file replication control method according to claim 14, wherein the other node that has the update permission releases the update permission after an update that has a dependent relationship with the update performed at the other node has been propagated to all the nodes.

17. (previously presented) The file replication control method according to claim 14, wherein said method further comprises:
the other node that has updated the file asynchronously propagating an updated content to the other nodes; and
causing the other node that has updated the file to process an access request that takes place in the access requesting node while the updated content is being propagated.

18. (original) The file replication control method according to claim 17, wherein the updated content is reflected in such a manner that order thereof is assured.

19. (original) The file replication control method according to claim 18, wherein a dependency information that represents order of other updates to be propagated to the other node along with the updated content.

20. (previously presented) The file replication control method according to claim 19, wherein a node that has received the updated content to reflect the updated content on a file of the node itself after receiving a previous updated content based on the dependency information.

21. (previously presented) The file replication control method according to claim 14, wherein a propagation mode of an updated content is designated for each of at least one file.

22. (previously presented) The file replication control method according to claim 14, wherein a node to which an updated content is propagated is designated for each of at least one file.

23. (previously presented) The file replication control method according to claim 14, further comprising:

restoring data of a file of a newly joined node; and

operating a user program before data of the file is completely restored.

24. (previously presented) The file replication control method according to claim 23, wherein restored data is transmitted in such a manner that order of update requests for the file is assured.

25. (previously presented) The file replication control method according to claim 23, wherein the node asks another node that shares the file to perform a process for an access request for the file when the access request takes place in the node itself before data is completely restored.

26. (previously presented) The file replication control method according to claim 14, wherein a node that has performed a systematic stop in which nodes that share a file are synchronously stopped to store a systematic stop state and the node synchronously resumes a process for the file without restoring data of the file.

27. (previously presented) A file replication method for a system having a plurality of nodes connected to a network, files being distributed to the nodes, comprising:

causing a first node to request a token for accessing a file;

causing the first node to access the file when the first node has the latest data of the file and is able to obtain the token for accessing the file from another node having update permission for the file which is given to only one node at a time;

notifying the first node of a second node that has the token when the first node is not capable of acquiring the token; and

causing the first node to ask the second node to access the file when the first node is notified that the first node is not capable of acquiring the token.

28. (previously presented) A computer-readable portable storage medium, when being used by a computer that composes a node connected to other nodes through a network in a file replication system, on which is recorded a program for causing the computer to execute a process, said process comprising:

when the node accesses a file and a node itself has the latest data of the file and has or is able to obtain access permission from another node having update permission for the file, causing the node itself to access the file of the node itself; and

when another node has the update permission for the file which is given to only one node at a time, causing the node itself to ask the other node to access the file.

29. (cancelled)